CLAIMS

1. A method of converting geometry into driving voltages for individual mirrors in a micromirror array, including

mapping a work piece into overlap regions, corresponding to patterns of intended energy contributions from a plurality of overlapping exposures of regions of the work piece;

rendering many polygons to gray scaled pixel values using a plurality of rendering engines operating in parallel;

calculating exposure values for the gray scaled pixel values, taking into account the overlap regions;

contouring the exposure values based on individual micromirror characteristics; and

outputting the contoured exposure values to one or more mirror pixel drivers.

2. The method of claim 1, wherein the mapping step produces 81 overlap regions in the irradiated area in one step.

3. The method of claim 1, wherein the calculating step uses the overlap regions to take into account overlap between successive exposures.

4. The method of claim 1, wherein the calculating step uses the overlap regions to take into account irradiation energy applied in successive exposures.

5. The method of claim 3, wherein the calculating step uses the overlap regions to take into account irradiation energy applied in successive exposures.

6. The method of claim 1, further including buffering the output of the rendering, calculating and contouring steps in at least one segmented buffer, wherein the outputting step includes repeatedly coupling one of the buffer segments to a set of mirror pixel drivers.

7. The method of claim 6, wherein the mirror pixel drivers are digital to analog converters switchably coupled to individual mirrors of a micromirror array.

8. The method of claim 6, wherein the segmented buffer accepts input from a number of rendering engines different from the number of buffer segments.

9. The method of claim 1, further including buffering the output of the rendering, calculating and contouring steps in at least two segmented buffers, wherein the outputting step includes repeatedly coupling one of the buffer segments to a set of mirror pixel drivers.

10. The method of claim 9, wherein the mirror pixel drivers are digital to analog converters switchably coupled to individual mirrors of a micromirror array.

11. The method of claim 9, wherein the segmented buffer accepts input from a number of rendering engines different from the number of buffer segments

12. A method of rendering, including:

providing a smaller addressing window, corresponding to a memory;

providing a larger addressing window, comprising the smaller addressing window and a guard zone outside the smaller addressing window, the guard zone being addressable but not corresponding to the memory;

receiving a polygon contained within the larger addressing window;

rendering at least sections of the polygon into the larger addressing window; and

writing data for the rendered sections of the polygon within the smaller addressing window to the memory and discarding said data for the rendered sections of the polygon outside the smaller addressing window.

13. The method of claim 12, wherein the rendering step is carried out without distinction between portions of the polygon inside and outside the smaller addressing window.

14. The method of claim 12, wherein the method is carried out without clipping the polygon to fit the smaller addressing window.

15. The method of claim 12, further including the step of filtering a set of polygons so that at least a portion of the polygon lies inside the smaller addressing window.

16. The method of claim 12, wherein the discarding said data is carried out by a filter before the writing the data for the rendered sections to a memory controller, thereby reducing memory bandwidth loads.

17. The method of claim 12, wherein the discarding said data is carried out by comparing memory address signals with valid addresses of the smaller addressing window and using a result of the comparison to control one or more memory arbitration signals.

18. The method of claim 13, further including the step of filtering a set of polygons so that at least a portion of the polygon lies inside the smaller addressing window.

19. The method of claim 14, further including the step of filtering a set of polygons so that at least a portion of the polygon lies inside the smaller addressing window.

20. A method of rendering polygons to a larger addressing window, the larger addressing window comprising a smaller addressing window corresponding to a memory and a guard zone outside the smaller addressing window, the guard zone being addressable but not corresponding to the memory, the method including:

receiving a polygon contained within the larger addressing window,

repeatedly selecting a section of the polygon;

converting the section into rasterized representation data;

processing the rasterized representation data to write portions of said rasterized representation data within the smaller addressing window to the memory, and not

to write portions of said rasterized representation data outside the smaller addressing window to the memory.

21. The method of claim 20, wherein the converting step is carried out without distinction between portions of the polygon inside and outside the smaller addressing window.

22. The method of claim 20, wherein the method is carried out without clipping the polygon to fit the smaller addressing window.

23. The method of claim 20, further including filtering the rasterized representation data before the writing to the memory, thereby reducing memory bandwidth loads.

24. The method of claim 20, further including comparing memory address signals with valid addresses of the smaller addressing window and using a result of the comparison to control one or more memory arbitration signals.

25. The method of claim 20, further including the step of filtering a set of polygons so that at least a portion of the polygon lies inside the smaller addressing window.

26. The method of claim 21, further including the step of filtering a set of polygons so that at least a portion of the polygon lies inside the smaller addressing window.

27. The method of claim 22, further including the step of filtering a set of polygons so that at least a portion of the polygon lies inside the smaller addressing window.

28. The method of claim 20, further including the step of decomposing a geometric feature into a set of polygons that are sized to fit within the guard zone.

29. The method of claim 21, further including the step of decomposing a geometric feature into a set of polygons that are sized to fit within the guard zone.

30. The method of claim 22, further including the step of decomposing a geometric feature into a set of polygons that are sized to fit within the guard zone.

31. A device to render a polygon contained within a larger addressing window, the larger addressing window comprising a smaller addressing window and a guard zone outside the smaller addressing window, the device including:

input lines;

a renderer connected to the input lines, adapted to receive a polygon contained within the larger addressing window and repeatedly to convert a section of the polygon into one or more addressing instructions and data;

a memory, coincident with the smaller addressing window;

a memory controller, connected to the renderer, adapted to process the addressing instructions and the data, to write portions of the data inside the smaller addressing window to the memory, and to disregard portions of the data outside the smaller addressing window.

32. The method of claim 31, wherein the renderer is adapted to convert the section without distinction between portions of the section inside and outside the smaller addressing window.

33. The method of claim 31, wherein the device does not clip the polygon to fit the smaller addressing window.

34. A device for writing to memory a section of a polygon contained within a larger addressing window, the larger addressing window comprising a smaller addressing window and a guard zone outside the smaller addressing window, the section represented by one or more addressing instructions and data, the device including:

input lines;

a memory coincident with the smaller addressing window;

a memory controller connected to the input lines and the memory, adapted to receive addressing instructions and data referencing the larger addressing window, to write portions of the data within the smaller addressing window to the memory, and to disregard portions of the data within the guard zone.

35. A method of representing an area of an image utilizing at least two levels of resolution, including:

receiving a polygon representation bounded by edges;

repeatedly selecting a section of the polygon for rendering, the section corresponding to pixels;

representing the pixels in a first data structure as filled, empty or partially filled, based on the edges of the polygon representation;

representing the partially filled pixels in a second data structure by arrays of sub-pixels that are filled or empty, defining the edges of the polygon representation, said arrays including at least 3 x 3 sub-pixels.

36. The method of claim 35, wherein the array further includes at least 4 x 4 sub-pixels.

37. The method of claim 35, wherein the array further includes at least 8 x 8 sub-pixels.

38. The method of claim 37, wherein the polygon is a quadrangle.

39. The method of claim 38, wherein the quadrangle is convex.

40. The method of claim 38, wherein one side of the quadrangle may have a zero length.

41. The method of claim 37, wherein the polygon is a trapezoid.

42. The method of claim 41, wherein one side of the trapezoid may have a zero length.

43. The method of claim 37, further including representing the filled and empty pixels as filled or empty.

44. The method of claim 37, further including representing the pixels as filled, empty or partially filled in two bits.

45. The method of claim 37, further including maintaining summaries of the partially filled pixels corresponding to filled or empty sub-pixels in the arrays.

46. The method of claim 45, wherein the summaries are counts of the sub-pixels in the arrays.

47. The method of claim 45, wherein the summaries are weighted evaluations of the sub-pixels in the arrays.

48. A method of representing an area in an image utilizing at least two levels of resolution, including:

receiving a polygon representation bounded by edges;

repeatedly selecting a section of the polygon for rendering, the section corresponding to pixels;

classifying the pixels as filled, empty or partially filled, based on the edges of the polygon representation;

representing partially filled pixels by arrays of sub-pixels that are assigned an intensity level, defining the edge of the polygon representation, said arrays including at least 3 x 3 sub-pixels.

49. The method of claim 48, wherein the array further includes at least 4 x 4 sub-pixels.

50. The method of claim 48, wherein the array further includes at least 8 x 8 sub-pixels.

51. The method of claim 50, wherein the polygon is a quadrangle.

52. The method of claim 51, wherein the quadrangle is convex.

53. The method of claim 51, wherein one side of the quadrangle may have a zero length.

54. The method of claim 50, wherein the polygon is a trapezoid.

55. The method of claim 54, wherein one side of the trapezoid may have a zero length.

56. The method of claim 50, further including representing the filled and empty pixels as on or off.

57. The method of claim 50, further including representing the pixels as filled, empty or partially filled in two bits.

58. The method of claim 50, further including maintaining summaries of the partially filled pixels corresponding to the intensity levels of the sub-pixels in the arrays.

59. The method of claim 58, wherein the summaries are summations of the sub-pixel intensity levels.

60. The method of claim 58, wherein the summaries are weighted evaluations of the sub-pixel intensity levels.

61. A data structure representing an area having at least one edge, utilizing two levels of resolution, including:

at least one memory;

at least one pixel map stored in the memory, representing a pixel as filled, empty or partially filled;

at least one sub-pixel array stored in the memory, corresponding to the pixel map, representing the partially filled pixel by an array of sub-pixels that are filled or empty, defining the edge of the area, said array including at least 3 x 3 sub-pixels, without representing the filled or empty pixel with sub-pixel values.

62. The data structure of claim 61, wherein the array further includes at least 4 x 4 sub-pixels.

63. The data structure of claim 61, wherein the array further includes at least 8 x 8 sub-pixels.

64. The method of claim 63, wherein the area is a quadrangle.

65. The method of claim 64, wherein the quadrangle is convex.

66. The method of claim 64, wherein one side of the quadrangle may have a zero length.

67. The method of claim 63, wherein the area is a trapezoid.

68. The method of claim 67, wherein one side of the trapezoid may have a zero length.

69. The data structure of claim 61, further including a gray value summary map summarizing filled or empty sub-pixels in the arrays.

70. The data structure of claim 69, wherein the summaries are counts of the sub-pixels in the arrays.

71. The data structure of claim 69, wherein the quantity is based on a weighted summation of the sub-pixels in the arrays.

72. The data structure of claim 61, wherein the sub-pixels are on, off or have an intermediate intensity level defining the edge of the area.

73. The data structure of claim 72, further including a gray value summary map summarizing the intensity levels of the sub-pixels in the arrays.

74. The data structure of claim 73, wherein the summary is a summation of the intensity levels of the sub-pixels in the arrays.

75. The data structure of claim 73, wherein the summary is a weighted summation of the intensity levels of the sub-pixels in the arrays.

76. The data structure of claim 61, wherein the pixel map and the sub-pixel arrays are stored in separately addressable memories.

77. The data structure of claim 61, wherein the pixel map and the gray value summary map are stored in one or more memories that are addressable separately from one or more memories in which the sub-pixel arrays are stored.

78. The data structure of claim 61, wherein a plurality of the filled pixels corresponding to a section bounded by opposing edges of the area is represented by a count of filled pixels.

79. The data structure of claim 62, wherein a plurality of the filled pixels corresponding to a section bounded by opposing edges of the area is represented by a count of filled pixels.

80. The data structure of claim 61, wherein particular pixels are represented by particular memory locations containing a value corresponding to filled, empty or partially filled.

81. The data structure of claim 62, wherein particular pixels are represented by particular memory locations containing a value corresponding to filled, empty or partially filled.

82. A data structure representing at least one trapezoid utilizing two levels of resolution, the trapezoid having first and third parallel edges that are parallel to each other and to a reference axis and having second and fourth opposing edges, including:

at least one memory;

at least one pixel map stored in the memory, representing pixels as filled, empty, parallel edge, or opposing edge;

at least one sub-pixel array stored in the memory, corresponding to the pixel map, representing the parallel edge pixel and the opposing edge pixel by an array of sub-pixels that are filled or empty, said array including at least format 3 x 3 sub-pixels.

83. The data structure of claim 82, wherein the array further includes at least 4 x 4 sub-pixels.

84. The data structure of claim 82, wherein the array further includes at least 8 x 8 sub-pixels.

85. The data structure of claim 82, wherein a plurality of first or third parallel edge pixels are represented by one array of sub-pixels.

86. The data structure of claim 82, further including a gray value summary map summarizing a quantity of sub-pixels that are filled or empty.

87. The data structure of claim 86, wherein the quantity is based on a count of sub-pixels that are filled or empty.

88. The data structure of claim 86, wherein the quantity is based on a weighted summation of sub-pixels that are filled or empty.

89. The data structure of claim 82, wherein the sub-pixels are on, off or have an intermediate intensity level defining the edge of the area.

90. The data structure of claim 89, wherein the quantity is based on a weighted summation of sub-pixels that are filled or empty.

91. The data structure of claim 89, wherein the quantity is based on a weighted summation of sub-pixels that are filled or empty.

92. The data structure of claim 82, wherein the pixel map and the sub-pixel array are stored in separately addressable memory.

93. The data structure of claim 82, wherein the adjacent parallel edge pixels are represented by the same sub-pixel array.

94. The data structure of claim 82, wherein the parallel edges can have zero length.

95. A protocol for transmitting graphic data representing a polygon having a plurality of edges, utilizing two levels of resolution, including:

representing a section of the polygon by an array of pixels, the pixels being assigned a value of filled, empty or partially filled;

representing the partially filled pixel by a sub-pixel array, the sub-pixels being assigned a value of filled or empty defining the edges, the array including at least 3 x 3 sub-pixels;

transmitting a representation of the array of pixels and a plurality of arrays of sub-pixels using at least a first channel and an independent second channel, the first

channel being used for the representation of the array of pixels and the second channel being used for the arrays of sub-pixels.

96. The protocol of claim 95, wherein the sub-pixel array further includes at least 4 x 4 sub-pixels.

97. The protocol of claim 95, wherein the sub-pixel array further includes at least 8 x 8 sub-pixels.

98. The protocol of claim 95, wherein the representation of the array of pixels is run length encoded.

99. The protocol of claim 95, wherein:

the convex polygon is a trapezoid;

the trapezoid has first and third parallel sides being parallel to a reference axis; and

a plurality of partially filled pixels corresponding the a first or third parallel side are represented by a single sub-pixel array.

100. The protocol of claim 95, further including a gray value summary map summarizing filled or empty sub-pixels in the array.

101. The data structure of claim 100, wherein the summary is a count of the sub-pixels in the array.

102. The data structure of claim 100, wherein the summary is based on a weighted evaluation of the sub-pixels in the array.

103. The protocol of claim 95, wherein the sub-pixels are on, off or have an intermediate intensity level defining the edge of the convex polygon.

104. The data structure of claim 100, wherein the quantity is based on a weighted summation of sub-pixels that are filled or empty.

105. The data structure of claim 100, wherein the quantity is based on a count of sub-pixels that are filled or empty.

106. The protocol of claim 95, wherein the pixel map and the sub-pixel array are stored in separately addressable memory.

107. A method of calculating a value for a multi-valued pixel corresponding to part of an edge of a polygon, including:

providing an array of sub-pixels;

providing a set of pre-calculated sub-pixel bar maps corresponding to edges having particular orientations;

representing the part of the edge of the polygon by sub-pixels that are filled or empty, by applying the pre-calculated sub-pixel bar maps;

super-sampling a set of sub-pixels corresponding to a pixel and assigning a value to the pixel corresponding to the number of sub-pixels in the set that are either filled or empty.

108. The method of claim 107, wherein the pre-calculated sub-pixel bar map is represented as an array of fill bars; and applying the pre-calculated sub-pixel bar map further includes applying an offset value to the fill bars corresponding to an intersection of the edge and a boundary of an area represented by the array of sub-pixels.

109. A method of defining an edge of a polygon within an area having sides, the area being subdivided into sub-pixels, including:

providing a plurality of pre-calculated sub-pixel bar maps corresponding to potential intercepts and orientations of the polygon edge with the sides of the area;

wherein the potential intercepts are limited to discrete positions along edges of the area;

wherein the potential orientations are limited to orientations that connect the discrete positions;

determining two intercepts of the polygon edge with the edges of the area;

determining an orientation of the polygon edge; and

applying one of the pre-calculated sub-pixel bar maps corresponding to at least one of the two intercepts and the orientation.

110. The method of claim 109, further including super-sampling one or more sets of sub-pixels to which one or more of the pre-calculated sub-pixel bar maps have been applied.

111. The method of claim 109, wherein the first area is subdivided by no more than 256 sub-pixels and the discrete positions are limited to no more than 65 positions per sub-pixel.

112. The method of claim 109, wherein the area is subdivided by no more than 64 sub-pixels and the discrete positions are limited to no more than 33 positions per sub-pixel.

113. The method of claim 109, wherein the area is subdivided into no more than 32 by 16 sub-pixels and there are 17 discrete positions along an edge of the sub-pixel.

114. The method of claim 109, wherein the area is subdivided into no more than 16 by 8 sub-pixels and there are 9 discrete positions along an edge of the sub-pixel.

115. The method of claim 109, wherein the pre-calculated sub-pixel bar maps are limited to a set of the potential orientations forming a range of approximately 45 degrees and the pre-calculated sub-pixel bar maps are transformed to cover a range of approximately 180 degrees.

116. The method of claim 109, wherein the pre-calculated sub-pixel bar maps include a combination of the potential orientations forming a range of approximately 45 degrees from one of the potential intercepts across a range of the discrete positions along an edge of one pixel.

117. The method of claim 109, wherein the pre-calculated sub-pixel bar maps are limited to a set of the potential orientations forming a range of approximately 45 degrees from one of the potential intercepts across a range of the discrete positions along an edge of one pixel.

118. The method of claim 117, wherein the pre-calculated bar maps are transformed to cover a range of the potential orientations of approximately 180 degrees.

119. The method of claim 109, wherein the first and second sides are opposing.

120. The method of claim 109, wherein the first and second sides are adjacent.

121. A method of defining an edge of a polygon within an area having sides, the area being subdivided into sub-pixels, including:

providing a plurality of pre-calculated sub-pixel bar maps corresponding to potential intercepts and orientations of the polygon edge with the area;

wherein the potential intercepts are limited to discrete positions along the sides of the area;

wherein the potential orientations are limited to orientations that connect the discrete positions;

determining two intercepts of the polygon edge and an extension of the polygon edge with the edges of the rectangular area;

determining an orientation of the polygon edge; and

applying one of the pre-calculated sub-pixel bar maps corresponding to at least one of the two intercepts and the orientation.

122. A method of defining an edge of a polygon within a rectangular area, the rectangular area being subdivided into sub-pixels, including:

providing a plurality pre-calculated sub-pixel bar maps corresponding to potential intercepts of the polygon edge with the rectangular area;

wherein the potential intercepts are limited to discrete positions along edges of the rectangular area;

determining two intercepts of the polygon edge with the edges of the rectangular area; and

applying one of pre-calculated sub-pixel bar maps based on the two intercepts.

123. A method of defining an edge of a polygon within a rectangular area, the rectangular area being subdivided into sub-pixels, including:

providing a plurality of pre-calculated sub-pixel bar maps corresponding to potential intercepts of the polygon edge with the rectangular area;

wherein the potential intercepts are limited to discrete positions along edges of the rectangular area;

determining two intercepts of the polygon edge and an extension of the polygon edge with the edges of the rectangular area; and

applying one of the pre-calculated sub-pixel bar maps corresponding to the two intercepts.

124. The method of claim 123, further including super-sampling one or more sets of sub-pixels to which one or more of the pre-calculated sub-pixel bar maps have been applied.

125. The method of claim 123, wherein two opposing edges of the rectangular area are subdivided by no more than 64 sub-pixels and the discrete positions are limited to no more than 33 positions per sub-pixel.

126. The method of claim 123, wherein the rectangular area is subdivided into 32 by 16 sub-pixels and there are 17 discrete positions from along one edge of the sub-pixel.

127. The method of claim 123, wherein the rectangular area is subdivided into 16 by 8 sub-pixels and there are 9 discrete positions from along one edge of the sub-pixel.

128. The method of claim 123, wherein the pre-calculated sub-pixel bar maps are limited to a set of potential orientations between the discrete positions forming a range of approximately 45 degrees and the pre-calculated sub-pixel bar maps are translated to cover a range of approximately 180 degrees.

129. The method of claim 123, wherein the pre-calculated sub-pixel bar maps include a combination of potential orientations between the discrete positions forming a range of approximately 45 degrees with one of the potential intercepts across a range of the discrete positions along one edge of one sub-pixel.

130. The method of claim 123, wherein the pre-calculated sub-pixel bar maps are limited to a combination of potential orientations between the discrete positions forming a range of approximately 45 degrees with one of the potential intercepts across a range of the discrete positions along one edge of one sub-pixel.

131. The method of claim 130, wherein the pre-calculated sub-pixel bar maps are translated to cover a range of the potential orientations of approximately 180 degrees.

132. A method of selecting a sub-pixel bar map to represent an edge intersecting an area, the area having at least first and second edges, the area being subdivided into sub-pixels, the method including:

constructing a segment between intercepts along the first and second edges, the segment defining first and second regions of the area;

wherein the intercepts are limited to discrete positions along edges of the area;

forming candidate sets of sub-pixels to represent the first region, the sub-pixels in the set being completely or partially within the first region;

determining a variance between area coverage of the candidate subsets and area coverage of the first region;

evaluating corners formed by combining the candidate subsets with sub-pixel bar maps for other segments;

selecting from among the candidate subsets based on the determination of variance and the evaluation of corners; and

storing the selected subset in a memory.

133. The method of claim 132, further including:

repeating the method for a set of segments defining potential intercepts along the first and second edges; and

culling redundant subsets.

134. The method of claim 133, wherein the culling takes into account a maximum number of subsets to represent the set of segments defining potential intercepts.

135. The method of claim 133, wherein the culling takes into account a maximum acceptable error from using the subsets to represent the set of segments defining potential intercepts.

136. A method of pre-calculating sub-pixel bar maps for sub-pixels corresponding to pixels, to define part of a polygon edge, including:

providing potential first intercepts for a line corresponding to the polygon edge along an edge of one of the sub-pixels on a first side of a pixel pair;

providing potential second intercepts for the line along a second side of the pixel pair opposed to the first side;

wherein the potential first and second intercepts are limited to discrete positions;

providing segments connecting the first potential intercepts and the second potential intercepts spanning a predetermined range of angles;

for the segments, selecting sub-pixel bar maps to define regions bounded by the segments;

wherein the selection of the sub-pixel bar maps takes into account

137. The method of claim 136, wherein the range of angles is approximately 45 degrees.

138. The method of claim 136, wherein the range of angles is approximately 90 degrees.

139. The method of claim 136, further including providing translations to apply the sub-pixel maps to a range of potential orientations of the line spanning a range of approximately 180 degrees.

140. The method of claim 136, further including providing translations to apply the sub-pixel maps to a range of potential orientations of the line spanning a range of 360 degrees.

141. The method of claim 136, further including

142. A method of calculating a value of a pixel corresponding to a corner at an intersection of first and second edges of a polygon, including:

providing a memory including a first array of sub-pixels and a second array of sub-pixels, both first and second arrays corresponding to a pixel;

extending the first and second edges into first and second lines;

setting the sub-pixels of the first array to filled or empty, corresponding to a first region defined by the first line;

setting the sub-pixels of the second array to filled or empty, corresponding to a second region defined by the second line;

calculating an intersection of the first array and the second array;

super-sampling from the calculated intersection a set of sub-pixels corresponding to a pixel and assigning a value to the pixel corresponding to the sub-pixels.

143. The method of claim 142, wherein the arrays of sub-pixels correspond to rectangles and setting the sub-pixels corresponding to the regions further includes providing a plurality of pre-calculated sub-pixel bar maps corresponding to potential intercepts and potential orientations of the lines with the rectangles.

144. The method of claim 143, wherein the potential intercepts are limited to discrete positions along edges of the rectangles and the potential orientations are limited to orientations that connect the discrete positions.

145. The method of claim 143, wherein two opposing edges of the rectangles are subdivided by no more than 64 sub-pixels and the discrete positions are limited to no more than 33 positions per sub-pixel.

146. The method of claim 143, wherein the rectangles are subdivided into 32 by 16 sub-pixels and there are 17 discrete positions from along one edge of the sub-pixel.

The method of claim 143, wherein the rectangles are subdivided into 16 by 8 sub-pixels and there are 9 discrete positions from along one edge of the sub-pixel.

147. A method for loading segments of a micromirror array, including:

rendering a plurality of polygons, utilizing a plurality of rendering processors operating in parallel;

writing digital values corresponding to the rendered polygons to a buffer, the buffer having a plurality of regions corresponding to the plurality of rendering processors and having a plurality of segments;

repeatedly writing the digital values from one of the buffer segments to a set of digital to analog converters having analog outputs; and

repeatedly switching the analog outputs among the micromirror array segments.

148. The method of claim 147, wherein the digital values from the plurality of rendering processors are queued, as rendering cycles are completed by the rendering processors, for the writing to the buffer regions.

149. A method for loading segments of a micromirror array, including:

rendering a plurality of polygons, utilizing a plurality of rendering processors operating in parallel;

writing digital values corresponding to the rendered polygons to a first buffer of a plurality of buffers, the buffers having a plurality of regions corresponding to the plurality of rendering processors and having a plurality of segments;

repeatedly transferring the digital values from one of the segments of the first buffer to a set of digital to analog converters having analog outputs;

repeatedly switching the analog outputs among the micromirror array segments;

wherein the rendering and writing steps proceed, utilizing a second buffer, while the transferring and switching steps proceed, utilizing the first buffer.

150. The method of claim 149, wherein the digital values from the plurality of rendering processors are queued, as rendering cycles are completed by the rendering processors, for the writing to the buffer regions.

151. A method of adjusting a size of a geometric feature in a rasterized image, the feature having edges represented by gray scaled pixels, the method including:

receiving at least one size adjustment parameter;

accessing a memory that stores the rasterized image;

applying a convolution kernel to a neighborhood of pixels in the memory, a center pixel of the neighborhood including an edge, to classify the edge as horizontal, vertical or 45 degrees with respect to the neighborhood of pixels and to estimate the orientation of the edge;

modifying gray scale values of the pixels in the neighborhood to apply the size adjustment parameter consistent with the estimated orientation, without modifying the placement of the geometric figure in the rasterized image, including:

for a horizontal edge, sliding gray values of a horizontal row of central pixels of the neighborhood in a vertical direction, followed by filling up from the center pixel in a horizontal direction;

for a vertical edge, sliding gray values of a vertical row of central pixels of the neighborhood in the horizontal direction, followed by filling up from the center pixel in the vertical direction; and

for a 45 degree edge, translating the gray scale value of the center pixel into a distance between the edge and an inside corner of the center pixel, applying the

adjustment parameter and translating a resulting modified distance back to gray scale values of the center pixel and of neighboring pixels.

152. The method of claim 151, further including:

detecting a corner of the feature by applying a convolution kernel; and

modifying gray values of the pixels in the neighborhood of the detected corner to displace the detected corner anisotropically inwards or outwards from the feature.

153. A method of adjusting widths of geometric feature in a gray scale image stored in a memory, the method including:

detecting an edge of a feature in a gray scale image, the edge represented by gray scaled pixels;

estimating an orientation of the detected edge;

classifying the orientation of the edge, from classes including at least horizontal, vertical and 45 degree orientations;

displacing the detected edge anisotropically inwards or outwards from the feature, consistent with the classification of the orientation.

154. The method of claim 153, wherein the detecting and estimating steps include application of at least one convolution kernel to a neighborhood of pixels including the edge.

155. The method of claim 153, wherein the displacing step for the horizontal and vertical orientations includes:

displacing a row or column of central pixels in a convolution kernel in a vertical or horizontal direction, respectively, followed by

displacing a center pixel of the convolution kernel in a horizontal or vertical direction, respectively.

156. The method of claim 153, wherein the displacing step for the 45 degree orientation includes:

translating the gray scale value of a center pixel of a convolution kernel into a distance between the edge and an inside corner of the center pixel;

displacing the detected edge inwards or outwards from inside corner of the pixel; and

translating a resulting modified distance to gray scale values of the center pixel and of neighboring pixels.

157. The method of claim 153, further including:

detecting a corner of the feature; and

displacing the detected corner anisotropically inwards or outwards from the feature.

158. A method of compensating for energy variation in a multi-pass, stepped irradiation of a high contrast layer on a work piece, the method including:

mapping a work piece into overlap regions, corresponding to patterns of intended energy contributions in multiple passes of irradiation;

irradiating an area of the work piece with energy;

recording the energy applied to the area for the overlap regions in the area; and

adjusting irradiation of particular overlap regions of the area during subsequent exposure passes based on the recorded energy applied to the particular overlap regions.

159. The method of claim 158, wherein the energy applied to the area is measured during the irradiating step.

160. The method of claim 159, wherein the energy applied is measured by sensing output of a source of the energy.

161. The method of claim 159, wherein the energy applied is measured by sensing scattering of the energy back from the work piece.

162. The method of claim 158, wherein four exposure passes are used and at least 81 overlap regions in an area are irradiated at once.

163. The method of claim 158, wherein the energy applied is at least partially reflected from an energy source by a micromirror array to the work piece.

164. The method of claim 158, wherein the energy applied is filtered by an LCD array, between an energy source and the work piece.

165. A method of compensating for response differences among individual micromirrors in a micromirror array, including:

providing an array of compensation functions are associated with individual micromirrors in a micromirror array, wherein the compensation functions transfer illumination data values to driving voltages; and

for an individual micromirror, converting illumination data values to driving voltages by applying the compensation function corresponding to the individual micromirror.

166. The method of claim 165, further including selecting one array of the compensation functions from a plurality of arrays of the compensation functions based on a recent operating history of the micromirror array.